Programming 2
Tutorial and Practical 5

This week's tutorial and practical sessions review the concepts introduced during the object-oriented programming topics, and asks why things were done the way they were.

Reviewing the Courier Scenario

Looking back at the courier scenario (both van+truck, and van+truck+aircraft), there were a number of ways to design the same thing. Some of you may have found that in tutorial 4, a couple of the design ideas discussed in the previous week did not lend themselves to being extended or maintained.

Before the Aircraft

To see how a design can be conceived with maintainability in mind, let's look back to a few designs of the simpler van+truck scenario. On the next page, there are a number of ways in which the scenario could be modelled.

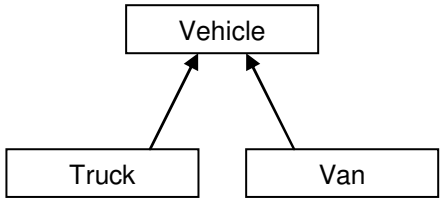
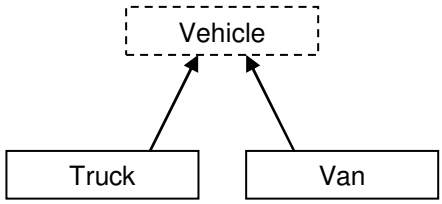
All these designs are a trade-off between a number of objectives.

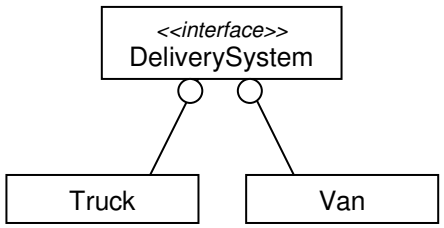
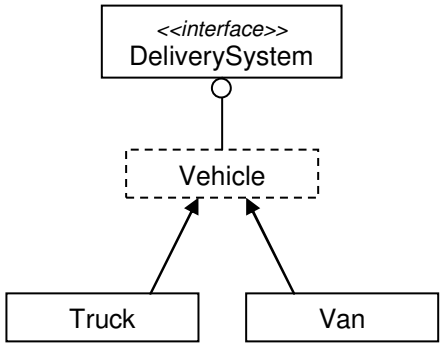
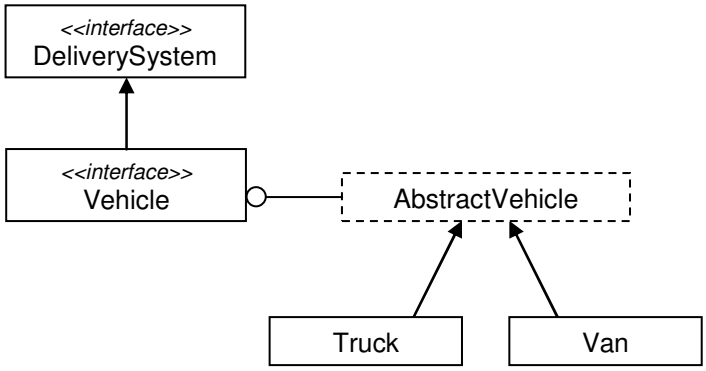
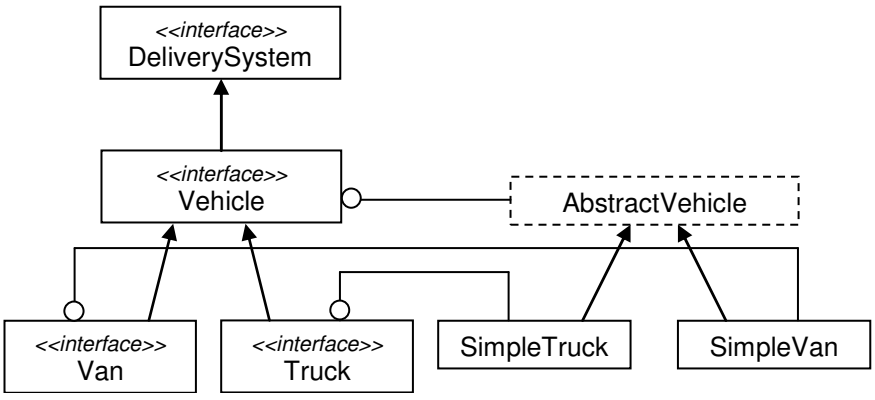
1. What are the advantages of each design?
2. What are the drawbacks of each design?
3. Identify the places where the design allows for extension. Discuss why these are made available, and how they might be useful.
4. In the case where the design is not extensible, why is it so? What problems occur?
5. When extended as in a) or b) below, do the classes remain cohesive?

When thinking of extending the scenario, consider both:

a) vehicle types (e.g. lorry, motorcycle, …)
b) non-vehicle types (e.g. aircraft, ship, train, …)

Design Proposals *(continued overleaf)*

A. Simple Inheritance Hierarchy	
B. Inheritance with Abstract Class	

C. Simple Interface Implementation	
D. Delivery Interface with Abstract Vehicle	
E. Delivery Interface With Parallel Vehicle Class and Interface	
F. Full Parallel Interface "Shadow"	

Practical Session

In the practical session, re-work your implementation from the previous practical session by adding the extra features from the updated scenario (i.e. adding a freighter aircraft to the system) covered in the previous tutorial.

Likewise, add (or remove) test cases as necessary from your testing program, to verify that the design does indeed work.